11 Discussing the DFRMS Architecture

11.1 Introduction

In the preceding chapter we presented the architecture for a DFRMS and provided detail on the important modules and components that the DFRMS comprises of. In this chapter we provide a more general discussion of the architecture and address why a DFRMS, as a system, is effective. The suitability of a DFRMS for particular organisations is also discussed. We look at potential weaknesses in access control and monitoring and how these can be mitigated. Furthermore, we show that a DFRMS can be used to help manage a digital FORCFIPI. In the final part of the chapter we present three scenarios that help demonstrate how a DFRMS can assist with DFR.

11.2 General Discussion

In each of the modules discussed in the previous chapter, the functionality of the modules meets or exceeds the requirements from the literature. Some functionality, such as monitoring, is obviously important. Other functionality, for example, storing the organisational hierarchy, may not seem as important; however, we recognise them all as being important for the following reason. The multiple individuals and departments – as well as the many interactions that occur between them in the course of DFR – can be considered a complex system. Whereas it may be possible to adequately manage each of these activities and individuals separately, we contend that the management of them all simultaneously is a difficult task prone to human error. Cook (2002, p.1) points out that "overt catastrophic failure occurs when small, apparently innocuous failures join to create opportunity for a systemic accident. Each of these small failures is necessary to cause catastrophe but only the combination is sufficient to permit failure". The DFRMS therefore assists management in ensuring that many small failures do not progress into a larger, more significant failure. In Section 11.5, we illustrate through scenarios how a DFRMS helps prevent large failures in managing DFR in a large organisation.

The implementation of a DFRMS, like the implementation of a digital FORCFIPI, is not a trivial undertaking. The setup of each module requires significant planning and time.

More important, however, is that the setup requires the input of non-DF staff. Such staff are likely to be from departments such as IT, IT security, information privacy and those departments whose business processes are the subject of DFR controls. In fact, staff from multiple departments will not only be needed for setup, but may also be needed in the daily use and regular maintenance of the DFRMS. This implies that for a DFRMS to be successfully implemented and used, it will require the buy-in of senior management from all the departments affected. The decision to implement a DFRMS therefore has to be made at a high level in the organisational hierarchy.

As mentioned previously, the DFRMS architecture is intended for use in a large organisation. It is also intended for an organisation with a well developed IT infrastructure, as well as mature IT security and DF programmes, such as a digital FORCFIPI. The absence of any of these would make the successful implementation and use of the DFRMS difficult. DF must also be of particular importance in an organisation to warrant the direct financial cost, as well as the cost in time and administrative overhead which the use of the DFRMS will entail. It is likely that a DFRMS may be suited for organisations that invest in the implementation of a digital FORCFIPI because the investment in a digital FORCFIPI is indicative of the importance of DFR in the organisation. A DFRMS is particularly suited to large organisations in industries where DF investigations are vital and may need to be undertaken as a matter of law. Such industries include the law enforcement, military and financial industries. The DFRMS's access control model in which trusted, high-level users may be monitored without their knowledge may be particularly useful in these industries where high-level users may need to be investigated.

The access control model in the DFRMS may blind users from the fact that they are being monitored; however, users with the sufficient knowledge and access to IT infrastructure may be able to infer that they are being monitored. Inferences may be made by intercepting network traffic, or indeed, where network traffic is encrypted, by observing network traffic patterns. Consider the case where a staff member's logins to an application server are being monitored. The staff member may notice unusual network traffic each time he logs into the server. To counter such analysis completely for trusted

staff may not be possible; however, certain measures may be taken to make it more difficult for the staff being monitored. First, communication between DFRMS components, as well as between devices and applications and the DFRMS should be encrypted where possible. Second, random traffic could be generated where feasible to reduce the effectiveness of traffic pattern analysis. Third, depending on the risk, certain components, modules, or even the DFMRS itself should be moved to an isolated segment of the organisation's network. The administration of this part of the network should be by DF staff only. The reason for isolating the DFRMS is that analysis may be performed on components such as the EAM's communications handler, which can expose the fact that specific monitoring is occurring through the alerts that are sent. Likewise, the activity in the communication links between the EAE and the alert store and event store databases may also disclose specific monitoring activity. Isolating the DFRMS or its components limits the visibility of such activity to DF staff only.

11.3 Integration with Existing Systems

In the discussion on the DFRMS thus far, we have not considered how a DFRMS may integrate with existing systems. A large organisation may posses a number of systems that contain similar data to a DFRMS or perform similar functions to a DFRMS. In order to avoid duplication, a DFRMS should ideally be able to interface with these systems. We consider integration with the following systems, which are typically found in large organisations: ERPs, SEMs, content management systems.

A DFMRS would need to interface with an ERP to retrieve financial information necessary to implement the Costing Module. Fortunately, ERPs have well defined interfaces for such interaction. Remote function calls, electronic data interchange (EDI), and application link enabling are examples of interface technologies that can be used. One popular ERP system has over ten such interfaces (Narayanan 2002, p.4).

The need to integrate with a SEM is that a DFRMS may receive security event information from a SEM. A SEM may serve the purpose of the EAM in a DFRMS. In order to receive event information from a SEM, a DFRMS would need the capability to use common security event standards, as mentioned in Section 10.4.1.

Content management systems can be used by a DFRMS to retrieve and store information that would be used in the DFRIMM. Like ERPs, content management systems also provide interfaces for integration. Some provide application programming interfaces (FileHold, 2012), while others provide software development kits (Computhink, 2012).

While integration is feasible from a technical point of view, an important factor to consider is access control. The DFRMS architecture proposed requires the ability to blind users to alerts and restricts different users from using certain functionality and accessing certain information. In order to ensure that the access control model is adhered to, the systems that are being interfaced with also need to be able to implement similar access control requirements. Besides being able to implement the DFRMS's access control model, administrative access to other systems is also a concern. Each of the systems discussed above have powerful administrator accounts. Users with access to these accounts may then be able to access or change information that is important for a DFRMS. For example, an administrator in a SEM may accidentally edit the events that a SEM will receive from a device. In the DFRMS, it is not possible to do this if an alert is based on the event. Thus, this control will be circumvented. An administrator of a content management system may also maliciously change a forensic procedure document.

Thus, there are two main barriers to integration: adherence to the DFRMS access control model and the access that will be available to multiple administrative users on the other systems. To solve these problems requires a deeper technical solution. We leave such a solution for further research.

Next we discuss how a DFRMS can be used together with a digital FORCFIPI.

11.4 Using a DFRMS with a Digital FORCFIPI

We have shown how the DFRMS assists in the management of DFR in general; however, we now narrow our discussion to the case of a DFR management framework, specifically the digital FORCFIPI discussed in Chapter 6. We explain how the DFRMS assists in the management of a digital FORCFIPI and how the DFRMS is able to include the

information privacy aspects of the digital FORCFIPI that may be over and above other DFR programmes.

Recall from Chapter 6 that the digital FORCFIPI consists of technical readiness procedures and processes as well as non-technical procedures and processes. The technical readiness procedures and processes in essence involve the configuration and monitoring of devices, as well as the auditing of such configuration and monitoring. In the prototype, the DFRIMM contains the procedures and processes for configuration, while the EAM ensures monitoring and logging.

Auditing functionality is not included in the DFRMS since the DFRMS assists in the operations of DFR. Auditing best practices espouse the principle of auditor independence, whereby the audit function is independent of the operations being audited (Elliott & Jacobson 1998). Thus, auditors should not be regular users of a DFRMS if they intend to audit the use of the DFRMS. The information held within the DFRMS, such as alerts, business processes and policies and procedures will, however, make auditing of DFR functions easier since auditors will have access to a central repository of DFR information when conducting audits. The benefits of this central repository of information may also accrue to information security and information privacy audits.

It is important, however, to consider information privacy when logging to a central repository. The information contained in the logs may violate the information privacy of employees. Accordingly, so-called privacy-respecting logging should be performed. This entails the encryption of log data, and that management provides access to the log data only to individuals who need it for specific purposes.

Purpose-based access control is an automated means of ensuring that access to information is restricted to the legitimate purposes of the user accessing the information (Yang et al., 2008). In order to maintain information privacy, it is essential that the use-limitation and purpose-binding FIPs are adhered to. Purpose-based access control helps ensure this. Given that the access control module of a DFRMS requires the use of interchangeable access control models, it is conceivable that a purpose-based access control model can be used to govern access to the DFRMS itself.

To consider instances where employees violate use limitation rules, it may be possible to monitor logs from applications and set alerts accordingly. For example, if a user accesses private information in an application and then emails a large attachment not long afterwards, he may fit the profile an employee that is emailing personal information out of the organisation. If a DFRMS is configured to receive the necessary events from the application and email server, it can trigger an alert. Of course, the email server and application need to have the ability to communicate with the DFRMS in order for the DFRMS to signal an alert. Another possible way of taking information privacy into account is to have an organisation use a privacy obligation management system such as the one proposed by Casassa Mont (2004). This system provides an interface between an organisation's users and its systems and data. Through the interface and underlying obligation management technology it aims to ensure information privacy is not violated. If a DFRMS can receive events from such a system, it will become far easier to detect information privacy violations.

The non-technical procedures and processes in the digital FORCFIPI consist of: internal forensic processes; the monitoring of business processes, business policies and business structure; and, the auditing of the aforementioned. Auditing is not included in the DFRMS for reasons mentioned above. Internal forensic processes in the digital FORCFIPI include education and certification – the management of these is assisted by the training management functionality in the DFRIMM. As with the digital FORCFIPI, which required the business or organisational structure to be stored and maintained, the architecture also requires the organisational structure to be stored within the DFRIMM. Business processes in the digital FORCFIPI are divided into privacy-related and privacy-specific business processes – both can be represented or stored in the DFRIMM since the DFRIMM is required to have the functionality to store business processes. By maintaining privacy-related and privacy-specific business processes, and in raising alerts related to these business processes, the DFRMS is able to cater for the information privacy requirements in the digital FORCFIPI.

In the discussion thus far, we have matched each area of the digital FORCFIPI with DFRMS functionality that can assist in the area. It is clear therefore that the DFRMS is

able to assist in the management of a digital FORCFIPI. If a DFRMS is used to manage a digital FORCFIPI, a DFRMS can be considered a PET by the definition presented in Section 2.3.5. Again, using the classification presented in Section 2.3.5, a DFRMS can be considered a high-level organisational PET, or HLO PET, when used in conjunction with a digital FORCFIPI. The DFRMS allows for the many functions required for DFR in a digital FORCFIPI to be contained in a single system rather than in many disparate systems.

In the following section, we look at three hypothetical scenarios in which a DFRMS based on the architecture presented in the previous chapter can be useful.

11.5 Scenarios

The scenarios detailed in this section are hypothetical in that they were not developed from knowledge of specific incidents at any organisation. They are, however, sufficiently generic for the purposes of example scenarios.

11.5.1 Scenario 1

This scenario involves a newly installed enterprise resource planning (ERP) server. The ERP server includes financial information and is used for, amongst other things, paying suppliers. In the scenario the newly installed server is missing an operating system (OS) patch. Employee X, an employee in the IT Department is paid by a foreign supplier to increase the price of the supplier's goods as listed on the ERP server. This is done because payments are made automatically to the supplier using price data from the ERP server. Employee X exploits the vulnerability exposed by the missing patch on the ERP server and gains access to the ERP server. He then proceeds to increase the price of the supplier's goods. A month later, having not been detected, Employee X resigns from his job and emigrates to a non-extradition treaty country.

Two weeks after Employee X resigns an audit detects the changed supplier prices, however the suppliers have already been paid, and since they are a foreign supplier recovery of the overpayment is not feasible. The organisation's DF team is notified, and while they suspect Employee X, they are unable to confirm this suspicion since the ERP

server operating system and the ERP software itself were not configured to log activity appropriately. No monitoring of event data from the ERP server OS or the ERP software itself was in place either. The organisation is therefore uncertain if Employee X was the perpetrator or if the perpetrator remains within the employ of the organisation.

A primary reason that Employee X was able to access the ERP server was the lack of timely patching of the operating system. Though ensuring that operating system patches are up to date is not a DF function, DF investigations often occur because of the failure of IT security controls or the lack of such controls. Thus, the missing patch is excluded from the scope of our analysis in this scenario. A number of failures then occur in this scenario that may have been avoided through a DFRMS built using the architecture proposed in the previous chapter. They are:

1. The ERP server OS was not configured to log appropriate user or network activity. This was because in the scenario no DF procedure for configuring the OS existed for the ERP server OS. A DFRMS would require that a procedure be specified at the time of adding the server to the DFRMS.
2. The ERP software itself was not configured for appropriate logging. Although supplier pricing was considered a risk area and a DF configuration procedure for the ERP was supplied by an external consulting firm, no DF personnel were sufficiently trained in the use of the ERP system to execute the configuration procedure. A DFRMS would require the necessary training be specified when adding the ERP to the DFRMS, thus alerting DF management to any risks due to lack of training.
3. No monitoring of event data from the ERP server OS or the ERP software itself was in place. When adding the server to the DFRMS event data would be a required parameter. While it can be specified in a DFRMS that no event data is necessary, the requirement for the event data forces management to consider whether such event data needs to be captured or not. In this scenario, if management was duly diligent it would have been able to configure monitoring by the DFRMS.

Each of the failures listed above contributed to the failure to be forensically ready investigate the incident. This reiterates the principle stated by Cook in the previous section, namely that a large failure is usually the culmination of a series of smaller failures.

11.5.2 Scenario 2

The second scenario also involves financial data. In this scenario, Manager Y, a financial manager has been defrauding the organisation she works for. Manager Y has managed to keep her activities undetected but realises that she may be caught out due to historical data stored in a financial application. Fearing an impending financial audit may uncover this data, Manager Y accesses the financial application late at night and begins to execute a large number of changes to cover her misdeeds.

The organisation, however, has security event management (SEM) software in place which has been configured to detect unusual activity in the financial application. Since Manager Y is executing a large number of transactions on the financial application and the transactions are occurring at an odd time, the SEM software raises an alert. The alert is seen by an IT security officer tasked with monitoring the SEM.

The IT security officer searches through the organisation's intranet and decides to call Manager Y since Manager Y is the manager responsible for the financial application. Manager Y tells the IT security officer that her department is performing tests on the data in the application and that is the reason for the large volumes of transactions and the unusual time the transactions are being executed. Not knowing better, the IT security officer does not report the alert and continues with his work. Manager Y successfully covers her tracks and is not detected during the financial audit.

In this case a DF investigation was not conducted as the correct escalation procedure was not followed. The appropriate procedure was for the IT security officer to escalate the incident to both the financial manager, Manager Y, and to the forensics department. This is because the organisation's policy requires the involvement of the forensics department for all incidents that affect certain business processes – the application in question is used in such a business process. The forensics department is then required to conduct a

preliminary investigation into the incident, in this case, to corroborate Manager Y's explanation. The IT security officer, however, did not have immediate access to the escalation procedure and used his initiative to contact Manager Y. Had a DFRMS been in place, the security officer would have been able to access the correct escalation procedure from the DFRMS's DFRIMM. He would have then involved the forensics department as soon as the incident was detected. A preliminary investigation by the forensics department would have been more likely to uncover Manager Y's fraudulent activity.

The third and final scenario follows.

11.5.3 Scenario 3

In this scenario the spokesman for a law enforcement agency is accused of leaking private medical information about the family of the head of the agency to the media. The spokesman claims that the information was obtained through a hack of his agency-issued smartphone. The agency makes use of a custom-developed smartphone application for communication since the application uses strong encryption. The spokesman, however, claims the leaked information was communicated to him via the custom-developed application. The internal affairs unit of the agency, which investigates agency staff, begins an investigation. Internal affairs requests their own DF unit examine the spokesman's smartphone to determine if it was in fact hacked. The internal affairs DF unit staff lack the skills to examine the smartphone and enlist the help of external DF consultants. After two weeks, the external consultants and internal affairs determine that the phone was indeed hacked and manage to trace the hack to a cellular modem that had been on the network for a week after the hack. A more timely response may therefore have allowed for the apprehension of the attacker.

The slow response by the law enforcement agency can be attributed to three factors: 1) the failure to link the use of the smartphone to the public relations business process, which involved communication with the spokesman; 2) the lack of a DF procedure to examine the smartphone; and 3) insufficient training of internal affairs' own DF unit in dealing with smartphones. Had a DFRMS been in use by the law enforcement agency,

this would have forced the agency to map its business process and realise that the smartphone is used in the public relations business process. The smartphone would then have been added to the DFRMS. The DFRMS would have also required a DF procedure for the smartphone when it was added, as well as appropriate DF training on the smartphone.

Each of the scenarios above provides examples of the potential of a DFRMS to aid in the management of DFR. The scenarios are specific in the area of DFR they address but illustrate the general point that a DFRMS has application in the management of DFR in general.

Next, we conclude the chapter.

11.6 Conclusion

In this short chapter we discussed the DFRMS architecture in general. We stated that organisations in which digital forensics is of particular importance make better candidates for DFRMS implementations. Likewise, organisations in which high-level staff may need to be monitored covertly can also benefit from a DFRMS. We mentioned weaknesses in access control and monitoring and how these can be overcome through isolating individual components or all of the DFRMS in a separate network segment. We also showed in the discussion that a DFRMS can be used in conjunction with a digital FORCFIPI. Finally we presented three scenarios designed to illustrate when a DFRMS can be useful.

In the next two chapters we discuss a DFRMS prototype developed according to the architecture presented in the previous two chapters.

12 DFRMS Prototype – The Event Analysis Module

12.1 Introduction

In order to test the concept of DFRMS, a proof-of-concept prototype of a DFRMS was developed based on the architecture presented in Chapter 10. The prototype, being a proof-of-concept-system, was not designed to be deployed in a large organisation. Rather, the focus when designing the prototype was to develop a system in which the basic or core functionality could be attained. The rationale for concentrating on the core functionality was that it would allow for a clearer understanding of how the various components functioned and how they affected one another before a larger, more capable system was attempted. Although at an early stage, the prototype was able to achieve the major functionality required of a DFRMS according to the architecture.

The prototype was developed using the Java programming language and MySQL database and tested on Windows XP and Windows 7 computers. While still a proof-of-concept DFRMS, the DFRMS developed was nevertheless sizeable, consisting of approximately 35 source packages and 225 individual classes. The discussion of the prototype is split between this chapter and the chapter that follows. In this chapter we describe how the prototype implements the architecture's event analysis module (EAM). In the chapter that follows we focus on the digital forensic readiness information management module and the other modules. Before describing the EAM though, we show the login screen in Figure 30 below. All users must access the prototype by entering the correct access credentials, namely a user name and password, at the login screen.

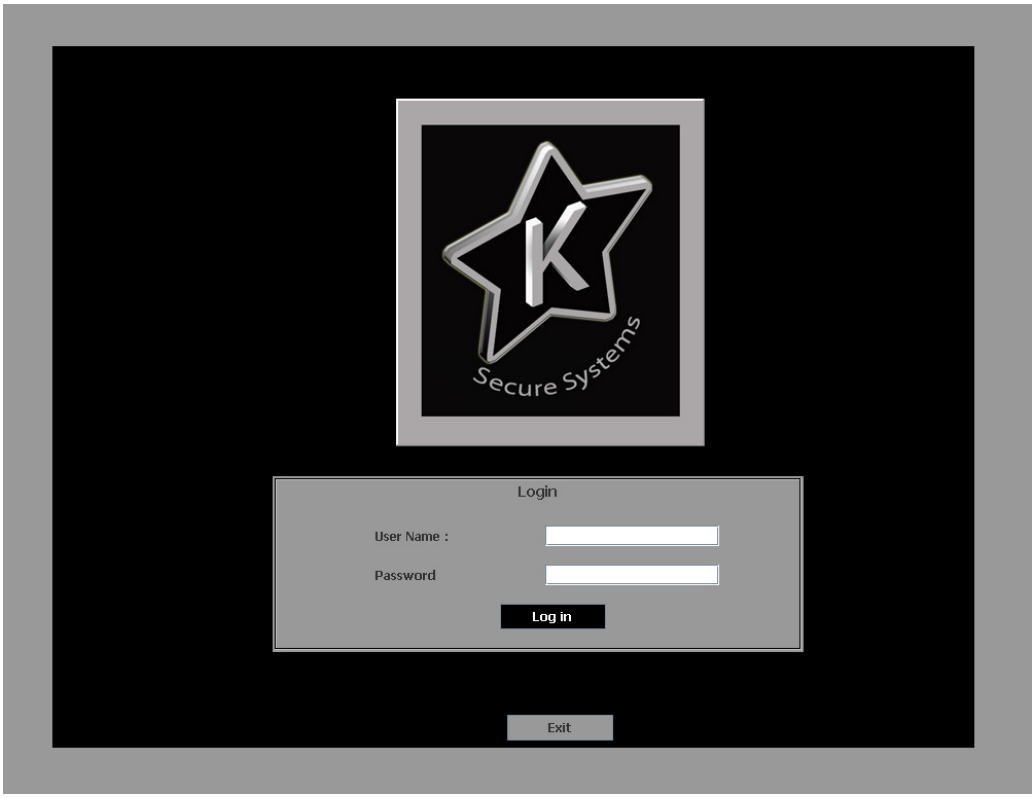


Figure 30 – Screenshot of login screen

Once logged in, users are faced with a welcome or home screen that shows the various modules and allows the users to select a module to work with. The home screen can be seen in Figure 31 below. The titles 'Monitoring', 'Information' and 'Costing' in Figure 31 refer to the EAM, DFRIMM and costing modules in the architecture, respectively. Users are also presented with a photograph of themselves and their *UserID*, or user identity, which is a unique numeric identifier assigned to all users by the DFRMS. The user's user name, email address, telephone number and rank in the organisational hierarchy are also displayed. Lastly, three warning indicators are displayed, namely 'Devices', 'Systems' and 'Alerts'. The meaning of these indicators is discussed in the sections that follow. In Figure 31 below, the photograph of the user has been deliberately pixelated for privacy reasons.

Figure 31 – Home or welcome screen

In the following section we discuss the EAM.

12.2 Event Analysis Module

The functions of the event analysis module (EAM) are accessed by clicking the 'Monitoring' button on the home screen, as shown in Figure 31. From the initial EAM screen, shown in Figure 32 below, users with appropriate access rights are able to:

- Create, modify or delete alerts
- View or delete event logs
- View or delete user activity logs
- Add, edit or delete devices or systems and their associated training requirements and forensic procedures

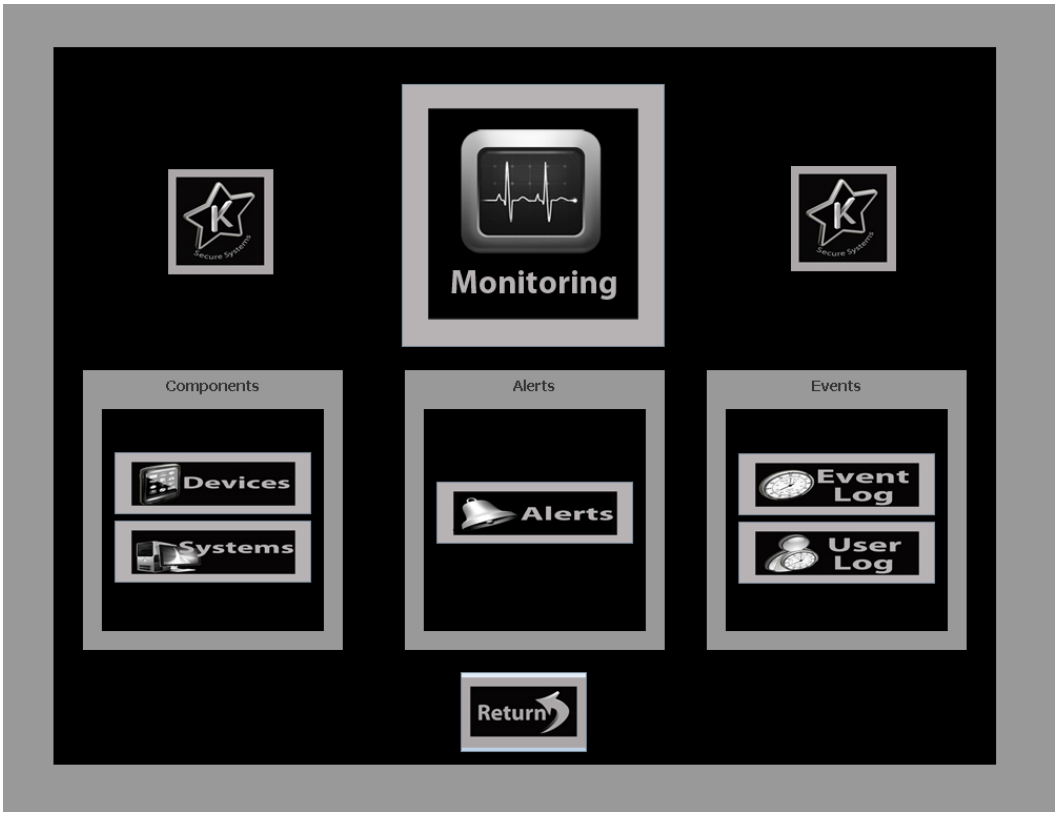


Figure 32 – Initial EAM or 'monitoring' screen

In the DFRMS devices are considered to be hardware entities capable of sending events to the EAM. Devices may include firewalls, routers, and even fingerprint readers. Systems, on the other hand are considered to be software entities, such as operating systems, application systems, database management systems, or any software capable of sending events. We begin by discussing the alert functionality in the sub-section that follows.

12.2.1 Alerts

As mentioned in the bullet list above, alerts can be created, modified or deleted by users with sufficient rights. Three levels of alerts are possible in the DFRMS, namely:

- **Critical.** Critical alerts are deemed to be of extreme importance. All users are automatically subscribed to critical alerts and, as such, it is not possible to blind specific users from these alerts. To avoid accidental deletion of critical alerts, the alerts must first be modified to a lower level and then deleted.

- **Medium.** These alerts are deemed important, however, they differ from Critical alerts in that: they are subscribed to by individual users, it is possible to blind specific users in the alert definition, and they can be deleted directly.
- **Low.** Low alerts represent low priority events or events that are predominantly informational by nature.

Figure 33 shows part of an alert definition. The alert has an ID, which is a unique identifier assigned to each alert by the DFRMS. The alert also has a name, which is provided by the user when it is defined. The level of the alert can be seen in Figure 33 and it too is chosen by the user when the alert is defined. The top-most image on the left is the image of the user that created the alert. Since the alert is a medium level alert, it is possible to blind users. Images of the blinded users are shown in the lower pane. A user-provided description of the alert is also included.

Figure 33 – Screen showing part of an alert definition for a medium alert

Alerts are defined with respect to events that are sent from devices and systems the DFRMS is configured to monitor. For example, an alert might be triggered if three

separate events, event A, event B and event C occur in a specific order. In Figure 33, these events are called 'conditions'. Clicking on the black button labelled 'Conditions' allows the user to select events in the appropriate order to trigger the alert – this can be seen in Figure 34.

To use a more specific example of an alert definition, consider Scenario 1 in the previous chapter. Assume that once Employee X has exploited the missing OS patch on the ERP server he is able to escalate the privilege of an OS service account to that of an administrator account. Assume further that Employee X uses this service account to make direct changes to ERP database files. Now, the DFRMS can be configured to trigger an alert when the ERP server OS sends the DFRMS an event signalling the account privilege escalation. To accomplish this in the DFRMS, the user clicks the 'Conditions' button shown in Figure 33, selects the appropriate system and then selects the system's events that will trigger the alert. Figure 34 shows the DFRMS once this has been done.

Figure 34 – Screenshot showing events or messages selected for alert definition.

In Figure 34, the 'Systems' tab has been selected. The system in question, namely the ERP server OS can be seen as the lower of the two systems displayed on the screen. Two of the messages or events that the DFRMS is configured to receive from the ERP server OS can also be seen. These events are highlighted in green, indicating that they have been selected by the user for the alert definition. The first event highlighted signifies an account privilege escalation, which means that an alert will trigger when this event is received by the DFRMS. The alert triggered in this case is the alert shown in Figure 33 since Figure 34 illustrates the events selected for that alert.

Although some SEM software uses sophisticated AI techniques to determine when to trigger an alert, these techniques were not the focus of the prototype. Instead, the prototype used simple pattern matching – if the events received matched a pattern associated with an alert, then that alert was triggered. The prototype was, however, designed according to the modular nature of the DFRMS architecture presented in Chapter 10. The simple pattern matching engine can therefore be easily replaced by a more sophisticated engine without significant change to the rest of the DFRMS.

The testing of alerts in the DFRMS was done through two separate programs. The programs simulated communication from devices and systems. Simulation was used for two reasons: first, to avoid the cost of acquiring systems and devices such as firewalls, routers, etc.; and second, to avoid the complexity and time required to write software to read and/or send events from such systems and devices. The first of the two programs developed for testing alerts allows a user to select a device or system. Once selected, any of the events the device or system is capable of sending can be selected and sent to the DFRMS. A screenshot of the testing program is shown below in Figure 35.

Figure 35 – Screenshot showing the alert testing program.

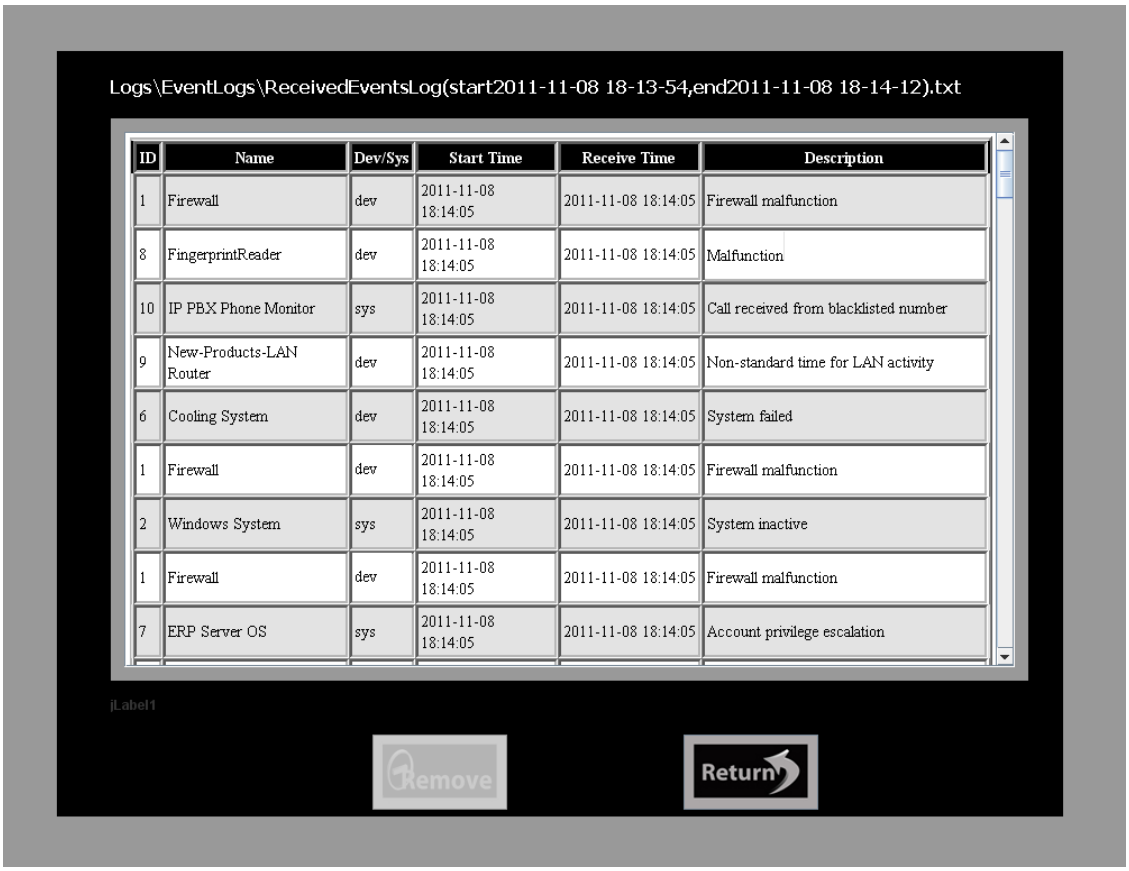
The second program for testing alerts randomly generates events from the devices and systems connected to the DFRMS. It does not have a graphical user interface and is hence not shown here.

In the next sub-section we discuss the logging capability of the EAM.

12.2.2 Event and User Logs

The DFRMS is typically configured to receive events from devices and systems connected to it. More detail on how this is done is covered later in the chapter in Section 12.2.3. Per the architecture in Chapter 10, all events received by the DFRMS should be stored. In the prototype all events are stored in encrypted event log files. These events are stored regardless of whether they are events that trigger alerts or not. Encryption is performed per the architecture in order to maintain the evidentiary value of the stored events. A weak encryption technique, namely the Caesar cipher, is used in the prototype. In a fully functional DFRMS, strong encryption would be used, together with an

appropriate key management system. Strong encryption is not used in the prototype as it was not a key focus area of the prototype. The prototype sought only to show that encryption could be used. Event logs may also be deleted by high-level users in order to save disk space or once the log files are no longer needed. Figure 36 below shows an event log from the DFRMS.



ID	Name	Dev/Sys	Start Time	Receive Time	Description
1	Firewall	dev	2011-11-08 18:14:05	2011-11-08 18:14:05	Firewall malfunction
8	FingerprintReader	dev	2011-11-08 18:14:05	2011-11-08 18:14:05	Malfunction
10	IP PBX Phone Monitor	sys	2011-11-08 18:14:05	2011-11-08 18:14:05	Call received from blacklisted number
9	New-Products-LAN Router	dev	2011-11-08 18:14:05	2011-11-08 18:14:05	Non-standard time for LAN activity
6	Cooling System	dev	2011-11-08 18:14:05	2011-11-08 18:14:05	System failed
1	Firewall	dev	2011-11-08 18:14:05	2011-11-08 18:14:05	Firewall malfunction
2	Windows System	sys	2011-11-08 18:14:05	2011-11-08 18:14:05	System inactive
1	Firewall	dev	2011-11-08 18:14:05	2011-11-08 18:14:05	Firewall malfunction
7	ERP Server OS	sys	2011-11-08 18:14:05	2011-11-08 18:14:05	Account privilege escalation

Figure 36 – Screen shot of an event log.

To explain the log information in Figure 36 we use the second line in the figure as an example. The '8' in the first columns refers to the 'ID' of the device or system from which the event emanated. As previously mentioned, the ID is a unique identifier given to the device or system by the DFRMS. The text 'FingerprintReader' in the second column is the name of the system or device that is given when the system or device is added to the DFRMS. In this example, 'FingerprintReader' refers to a biometric fingerprint access control device. The 'dev' in the second column indicates that it is a device and not a system which generated the event. A 'sys' would indicate a system-

generated event. The next column, labelled 'Start Time' is the timestamp of the event from the device or system. The subsequent column 'Receive Time' is the time the DFRMS received the event. These two times may not necessarily be the same since events may queue either at the device or system sending them, and/or in a queue at the DFMRS when it receives events. Further, delays or lags in network traffic may cause a difference in the start time and receive time. In Figure 36 the start and receive times are identical for each device since devices and systems were being simulated and the events were fired automatically from a separate program running on the same computer as the DFRMS. The final column contains a description of the event as entered by the user when configuring the device or system. In the example, the description 'Malfunction' indicates that the finger print reader has malfunctioned. Finally, the 'Remove' button on the bottom-left of Figure 36 allows a user with sufficient privileges to delete the log file. The button is greyed out in Figure 36 as the user had insufficient privileges to delete the log file.

The activity of users while using the DFRMS is also logged and stored in user logs. The DFRMS accumulates 300 user events before writing those events to a log file in encrypted form. This is, however, a flaw in the prototype implementation of the DFRMS since a failure in the DFRMS may result in events in the buffer not being written to logs.

12.2.3 Devices and Systems

In this sub-section we discuss how devices and systems are added, updated and deleted within the DFRMS, as well how the DFRMS is configured to receive events from devices and systems. As mentioned in Section 12.2.1 above, real devices and systems were not used. Instead devices and systems were simulated by separate programs which sent events to the DFRMS.

Figure 37 – Screenshot of initial screen for adding a device.

The addition of devices or systems is possible by pressing the 'Devices' or 'Systems' buttons in the initial EAM or 'monitoring' screen shown in Figure 32. Although each button loads a different screen, the procedure to add a device or system is the same. The procedure is as follows. First, the name of the device or system must be entered by the user. Second, the user presses the 'Device messages' or 'System messages' to go to another screen in which the user can specify the events sent by the device or system. Additional screens will be accessed if the user clicks on the 'Procedure' field or 'Training Requirements' area. These additional screens allow the user to associate a forensic procedure and training requirements with the device or system. We discuss these additional screens further in the next chapter which discusses the DFRIMM.

Figure 37 shows the first screen for adding a device – the screen for adding a system is almost identical. The black 'Device messages' button can clearly be seen in Figure 37. Figure 38 below shows the second screen for configuring a device or system, which is accessed by pressing the 'Device messages' button. In this screen the user enters the

name of the event that will be sent by the device or system. This method of associating events with devices and systems is only adopted since the prototype is for proof-of-concept purposes. In a fully-functional DFRMS that is used in a large organisation, it is most likely that the vendor supplying the DFRMS would provide the event types for each device with the DFRMS. The vendor would also supply subsequent updates for new or changed event types.

The number on the left of the text field is a unique identifier assigned by the DFRMS for events sent by the device or system. The 'remove' button on the right allows a user with sufficient privileges to remove or delete the event. In order to prevent users from stopping alerts from being triggered, it is not possible to delete an event if that event is needed to trigger an alert. The definition of the alert must first be changed such that it does not include the particular event before the event can be deleted.

Figure 38 – Screenshot of event definition screen.

The updating or deletion of devices and systems is simple to perform. When a device is selected for updating or deletion the user is presented with the same screens for adding a device, as shown in Figures 38 and 37. In the case of updating, the user is able to edit or change any of the details in the same manner as when the device or system was added. The only difference is that instead of a button for adding the device or system, the user is presented with buttons for updating or removing the system or device. Users must have sufficient privileges to successfully delete a device or system. Again, it is not possible to update or delete a device or system if the device or system is already part of an alert. Besides being a security precaution against alerts being circumvented, this also ensures that devices are properly configured before alerts are based on them.

In the section that follows we discuss functionality that was proposed in the architecture but not implemented in the EAM of the prototype.

12.3 Features Not Implemented

In the introduction to this chapter we stated that some features of the DFRMS architecture presented in Chapter 10 were not implemented in the prototype. The architecture presented in Chapter 10 is intended for a fully operational DFRMS within a large organisation, while the prototype was designed as a proof-of-concept system. The aim of the proof-of-concept system was to prove the basic concept by focusing on the core, or fundamental, functionality. The large size of the prototype (225 individual classes) and the limited amount of time available for development was also a factor in omitting non-core functionality, or functionality that could be easily added at a later stage in future work. The following two features were not implemented in the prototype's EAM:

- **Storage of alert definitions in encrypted form in the database.** This was not performed but is technically not difficult to implement since encryption and decryption libraries were used for event storage.
- **Alerts for business processes.** This involved raising alerts for the staff associated with business processes if a device or system in the business process

triggered an alert. Implementing the functionality would make use of current alert definitions and was therefore also not implemented.

The section that follows concludes this chapter.

12.4 Conclusion

In this chapter we introduced our proof-of-concept DFRMS prototype. The prototype served to prove that the concept of the DFRMS architecture presented in Chapter 10 could be implemented. The prototype is, however, discussed over two chapters, namely this chapter and the next chapter. In this chapter we described the implementation of the event analysis module or EAM. In particular, we discussed the functionality for alerts and the testing of alerts. Event and user logs, as well as the configuration of devices and systems, were also discussed in detail. The prototype showed that most of the EAM functionality dictated by the architecture could be implemented. Two features were not implemented: (1) the storage of alert definitions in encrypted form, and (2) alerts for business processes. These were not implemented primarily due to the limited amount of available time but may be implemented with little difficulty in future work.

In the following chapter we continue discussing our DFRMS prototype. We deal predominantly with the digital forensic readiness information management module but also include the remaining modules of the architecture.

13 DFRMS Prototype – Information, Access Control and User Interface Modules

13.1 Introduction

In the previous chapter our proof-of-concept DFRMS prototype was introduced and its EAM functionality discussed. In this chapter we continue discussing the prototype; however, we dedicate most of the discussion to the prototype's implementation of the digital forensic readiness information management module (DFRIMM). The implementations of the access control and user interface modules are also discussed. We begin with the DFRIMM.

13.2 Digital Forensic Readiness Information Management Module

The functions of the DFRIMM implemented in the prototype are accessed by clicking on the 'Information' button in the initial welcome or home screen, shown in Figure 31 in Section 12.1 in the previous chapter. Clicking on the 'Information' button brings the user to the initial DFRIMM screen, shown in Figure 39 below.

The options available to the user on the initial DFRIMM screen are grouped into three categories: 'General', 'Components' and 'Procedures'. These categories can be seen in Figure 39 below. Table 10 summarises the functionality available in the 'General' and 'Procedures' categories. The table lists the buttons available in each category and then lists the functionality that the button provides for.

Table 10 – Summary of options available from initial DFRIMM screen.

Category:	Button:	Allows For Administration of:
General	Users	Users in the DFRMS
	Teams	DF and incident response teams
	Training	DFR training needs
Procedures	Business Process	Business processes
	Docs	Policy and procedure documents
Components	As in EAM	

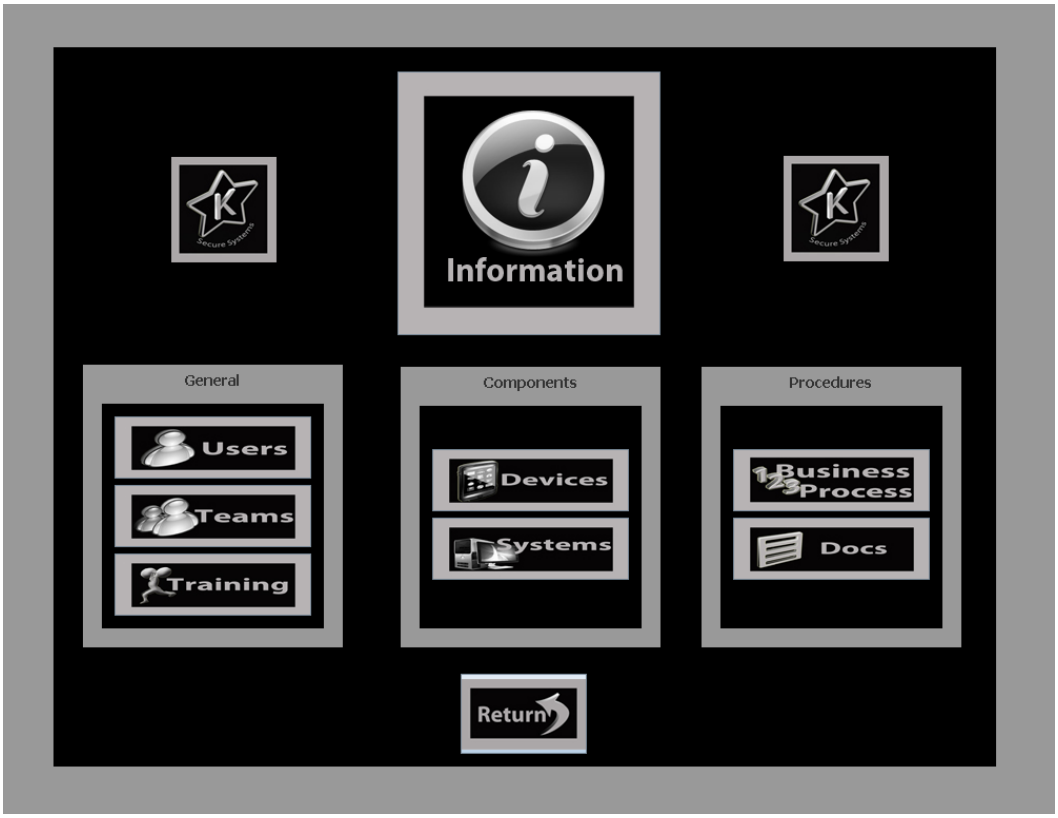


Figure 39 – Screenshot of initial DFRIMM or 'Information' screen.

The detail of the 'Components' category is not included in Table 10. This is because the 'Components' category provides for the management of devices and systems in exactly the same manner as the EAM, which has already been discussed in Section 12.2.3. The functionality is duplicated in the DFRIMM since devices may be added that are not part of any monitoring that is taking place. For instance, such devices or systems may be undergoing testing prior to implementation. Alternatively, the devices or systems may not be part of any monitoring activity, yet the organisation may find it prudent to maintain information about them in case they are involved in incidents or investigations. It should be noted, though, that devices or systems entered into the EAM are accessible via the DFRIMM. They do not have to be entered separately for the EAM and DFRIMM.

In the sub-sections that follow we expand on the summary presented in Table 10.

13.2.1 Users

The ability to administer DFRMS users is accessed by clicking the 'Users' button in the initial DFRIMM screen shown in Figure 39. Once the button is clicked, another screen is presented from which DFRMS users can be selected. DFRMS users can be deleted from the system, or their user details edited. New users may also be added to the system. All of the options for user administration that have been mentioned are only available to users with appropriate privileges.

Next, we discuss teams.

13.2.2 Teams

DF and incident response teams may be created, edited or deleted after clicking on the 'Teams' button in the initial DFRIMM screen. The 'Teams' button brings up another screen that displays a list of existing teams and which offers the option to add a new team. If a team is selected, the team information screen is presented to the user. This screen is shown in Figure 40 below.

Figure 40 – Screenshot of team information screen.

All of the information presented in Figure 40, with the exception of the team ID, is editable by clicking on the information. As with other IDs in the system, the team ID is a unique numeric identifier that is automatically generated. The team name is provided by the user when the team is created. A team icon is also selected by the user when the team is created. The icon provides for ease of reference when working with lists of teams. Photographs of the team owner and team members are also presented. The team owner is the user that created the team. When the team icon or team members are clicked, further screens are brought up to enable selection of an icon or team member, respectively. The 'Remove' button deletes the team from the DFRMS, while the 'Update' button commits any team information changes to the database.

When adding a new team, the user is presented with a screen almost identical to that shown in Figure 40. The user is merely required to enter or select the relevant information or items to create a new team.

Next, we discuss training.

13.2.3 Training

The 'Training' button in the initial DFRIMM screen shown in Figure 39 takes users to a screen that allows them to administer training courses associated with DFR. Once the 'Training' button has been pressed, the user is presented with the initial training menu. The initial training menu allows users to choose from the following list: 'Current Courses', 'Completed Courses' and 'All Courses'.

If the user selects 'Current Courses', the courses that the user is currently enrolled in are shown in a new screen. This is seen in Figure 41 below. The screen in Figure 41 shows the individual courses that the user is currently taking. Each course is displayed with a unique course code, for example, in the first line in Figure 41 'COS 222' is the course code. The course name is then listed, followed by the cost of the course. The total cost of current training courses for the user is also displayed at the top of the screen next to the user's photograph.

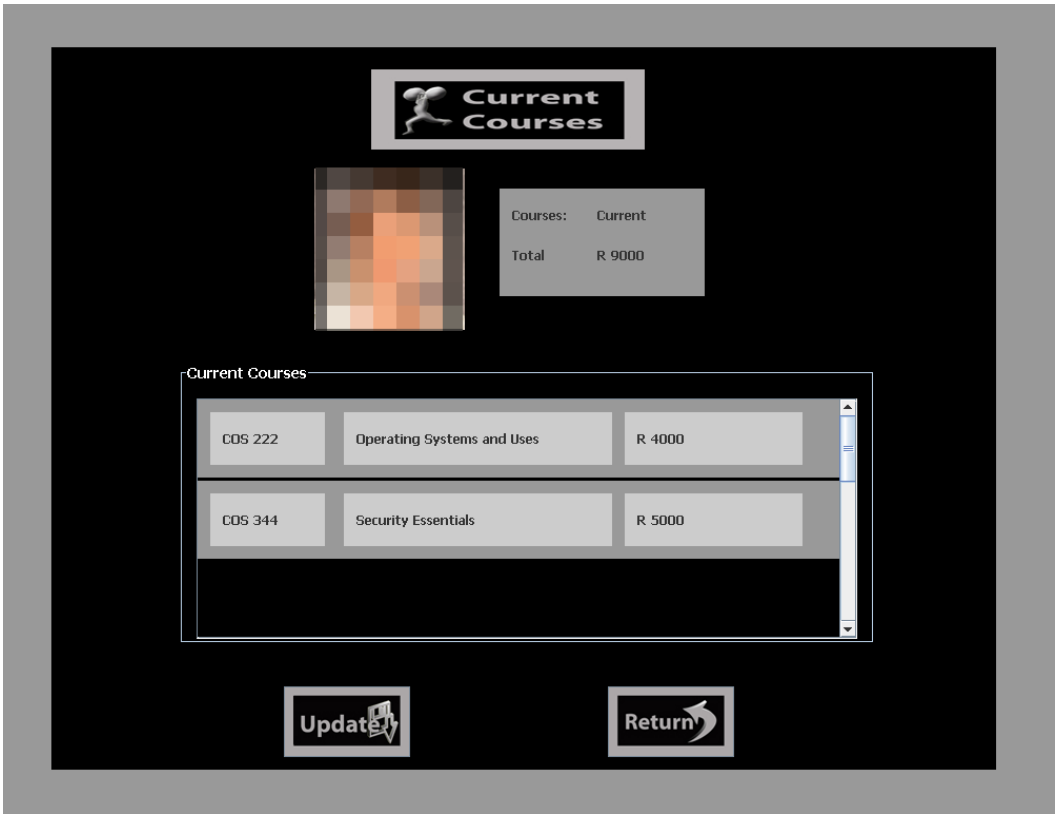


Figure 41 – Screenshot of screen showing current courses for a user.

If the user selects 'Completed Courses' from the initial training menu, the user will be presented with a screen similar to Figure 41, with the exception that the information presented is for training courses already completed.

The user may also select 'All Courses' from the initial training menu if the user has sufficient privileges. Clicking the 'All Courses' button takes the user to a new screen where the user can administer all the courses in the system. The user is able to add new courses, remove existing courses and change any detail about an existing course from the new screen.

An important part of the training functionality in the system is that, per the architecture, training courses should be associated with specific devices or systems. This is done when adding or updating devices or systems. Recall from the previous chapter, in Figure 37 in Section 12.2.3 when adding a device or system, the user may click in the 'Training Requirements' area to choose appropriate training courses for the device. When the user

clicks on the 'Training Requirements' area, a screen such as that shown in Figure 42 appears.

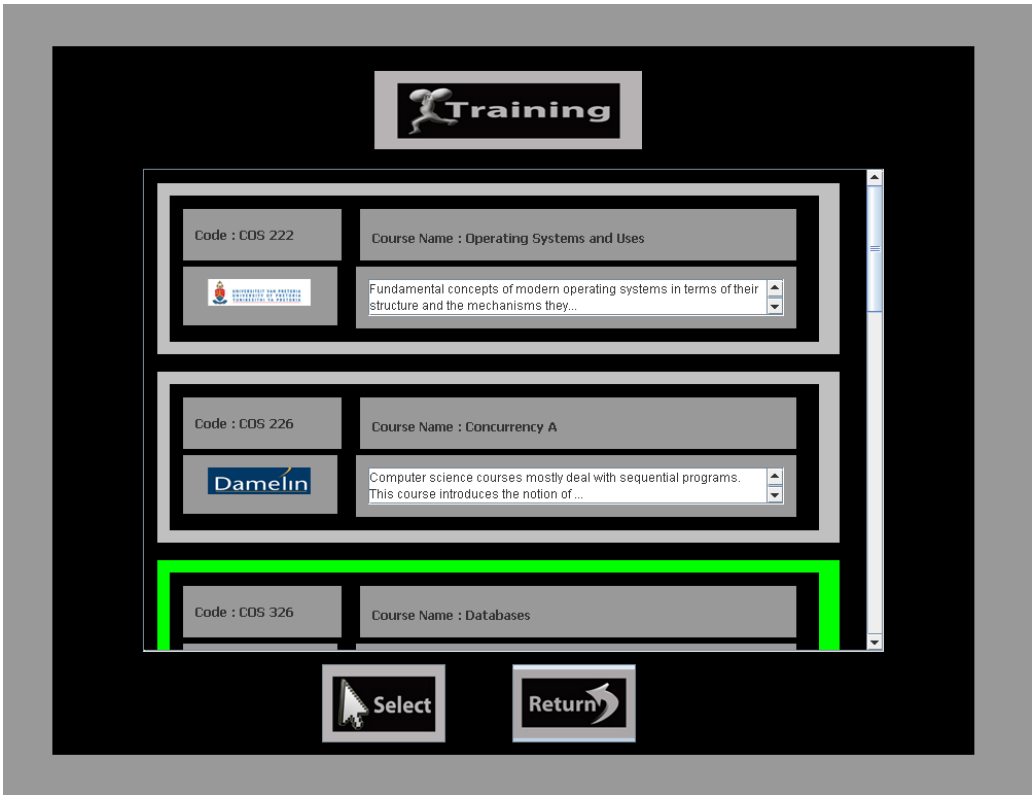


Figure 42 – Screenshot of all courses available for selection.

The course code, course name, a short description of the course and a graphic representing the institution offering the course are all displayed in the screenshot in Figure 42. The bottom-most course is highlighted in green indicating that it has been selected. The user may select these courses and they are then displayed as part of the training requirements for the device or system when it is added to the DFRMS.

In Section 12.1 of the previous chapter, Figure 31 shows the DFRMS home or welcome screen. Red warning indicators next to the words 'Devices' and 'Systems' are visible in Figure 31. These warning indicators signal the fact that devices or systems added by the current user do not have training requirements associated with them. In the scenario presented in Section 11.5.3 one of the flaws that prevented a more timely response of the smartphone hack was the lack of training by internal DF staff on smartphone forensics. Using the DFRMS, the lack of a training course for the smartphone would trigger the

warning indicator. The warning indicators mentioned also signal the lack of a forensic analysis procedure; however, this is discussed later in Section 13.2.5.

13.2.4 Business Processes

In this sub-section we discuss how the DFRMS includes information about business processes. Information regarding business processes is obtained by clicking the 'Business Processing' button in the initial DFRIMM screen shown in Figure 39. This brings up the initial business process screen, which shows all the business processes listed in the DFRMS. Each business process displayed in the initial business process screen is listed with its name, ID and a photograph of the business processes owner. The option to create a new business process is also presented.

If the user clicks on an existing business process, a new screen is displayed as shown in Figure 43.

Figure 43 – Screenshot of a screen showing an existing business process.

The name and description fields shown in Figure 43 are provided by the user when the business process is created. In Section 10.4.1 we used the following definition of a business process by Hammer and Champy cited in Lindsay et al. (2003, p.1015): a set of partially ordered activities intended to reach a goal. In the DFRMS these activities are called business activities. Three such business activities can be seen in Figure 43. Clicking on a business activity in Figure 43 brings up a further screen that shows each business activity and the staff that participate in the business activity. Right-clicking on a business activity in this screen again displays a new screen. The new screen provides a detailed description of the business activity, shows the participants and provides a button to allow for updating. It is shown in Figure 44.

Figure 44 – Screenshot showing the detail of a business activity.

Figure 44 displays information for the second business activity in the business process shown in Figure 43, namely the 'Initiate Investigation' business process. This business process is a fictional business process for initiating an investigation in the scenario presented in Section 11.5.3. In that scenario the internal affairs DF department did not

identify the smartphone as device of interest – that is, a device that may be part of an investigation. The scenario involved an investigation being improperly initiated through a smartphone application and the internal affairs department being unprepared to analyse the smartphone. The last line of the description in Figure 44 clearly shows that a smartphone application may be used in the business activity and by definition, the business process too. A DF department making use of the DFRMS can see the smartphone in the description. They can therefore ensure that the forensic analysis of smartphones is possible and avoid the state of unpreparedness that occurred in the scenario. Of course, relying on business process and activity information to identify important devices or systems that are in use implies that the business process and activity information must be accurate and complete.

The next sub-section deals with the last piece of functionality in the DFRIMM, namely the storage of DFR-related documentation.

13.2.5 Documentation

The DFRMS architecture presented in Chapter 10 requires that DFR-related policy and procedure documents are stored in the DFRMS for easy access in the event of an incident or investigation. In the prototype such documentation can be accessed by clicking the 'Docs' button in the initial DFRIMM screen shown in Figure 39. Once clicked, the 'Docs' button displays a new screen that has buttons for forensic and suspicion policies, as well as escalation and forensics procedures. The screen also has a button for adding new policies or procedures to the DFRMS. Clicking on any of the policy or procedure buttons brings up a list of all the respective policies or procedures in the DFRMS. For example, Figure 45 below was displayed by clicking the 'Escalation Proc.' button. Figure 45 shows a list of all the escalation procedures in the DFRMS.

As can be seen in Figure 45, the ID of each procedure document is displayed, together with the name of the document and a photograph of the user that added the document to the system. All policy and procedure documents are displayed in the same way as the escalation procedure documents in Figure 45.

Figure 45 – Screenshot showing list of escalation procedures in the DFRMS.

Clicking on a single policy or procedure document from a list like that in Figure 45 displays another screen which contains the contents of the document. The contents can be edited and saved by users with sufficient privileges. Users with sufficient privileges also have the option of deleting the document. In the scenario in Section 11.5.2, the lack of availability of an escalation procedure resulted in an incorrect procedure being followed and a crime going unnoticed. The use of a DFRMS such as the prototype would have allowed the IT security officer in the scenario to access the escalation procedure for financial application incidents. This procedure can be seen as the first in the list shown in Figure 45.

In Section 12.2.3 of the previous chapter we mentioned that forensic procedure documents can be associated with devices when adding devices or systems to the DFRMS. When forensic procedure documents are linked to devices or systems, it is not possible to delete the forensic procedure without first removing the procedure from the device or system's configuration. This is to stop users from deleting procedure

documents and creating a state of unpreparedness wherein the procedure is required and is not available. If a device or system lacks a forensic procedure document, a warning indicator is activated in the home or welcome screen, as shown in Figure 31 in Section 12.1.

13.3 Access Control and User Interface Modules

In this sub-section we first discuss the access control module and then user interface module of the prototype.

13.3.1 Access Control Module

The access control module (ACM) in the prototype implements a linear rank hierarchy. More specifically, the ACM implements an access control model based on a linear rank hierarchy. In the rank hierarchy, users belong to one of four ranks. These ranks are, in order of highest to lowest rank: Chief Forensic Officer, Forensic Officer, Security Officer and External Consultant. Each rank is represented by an integer and the higher the rank used, the greater the integer. The permission to access information or use functionality in the DFRMS also has a rank associated with it, which is known as the access rank.

When access to information or functionality in the DFRMS is required by a user, the ACM compares the rank of the user with the access rank. If the user's rank is greater or equal to the access rank, access is granted. This comparison occurs in a rank comparison method within the ACM. A 'pluggable' architecture is used in the ACM, which allows the rank comparison method to be overridden. If overridden, access comparisons may be performed in a different manner, for example, to deal with custom access control requirements.

The four ranks chosen for the prototype are unlikely to be adequate for an operational DFRMS used in a large organisation. The ranks were chosen for proof-of-concept purposes to illustrate the functionality of the ACM. Increasing the number of ranks is trivial, though time-consuming, which is the reason more ranks were not included. The rights associated with each rank in the prototype are listed in Appendix G.

13.3.2 User Interface Module

The user interface module (UIM) employed a graphical user interface (GUI). The GUI design in the prototype did not follow the more traditional, drop-down menu user interface. Instead, we opted for a user friendly icon-driven GUI based on the premise that this may improve user acceptance and ease of use of the DFRMS. As can be seen in the screen shots thus far, large icons and graphical elements featured extensively. In certain cases, however, the choice of GUI would not prove to be practical. For example, consider Figure 34 that displays events which form part of alerts. Only two alerts are visible per system. In a real-world setting, a system may contain hundreds of alerts, thus, the GUI would be impractical in such a case. We did not test our premise that the icon-driven GUI improves user acceptance and ease of use since it is out of the scope of this research. Testing would, however, determine if the GUI was better accepted and easier to use than a typical GUI, and in which situations this is the case. We leave the testing of the GUI as future work.

In accordance with the architecture presented in Chapter 10, the GUI module is not coupled with the EAM or DFRIMM. It is, however, dependent on the ACM since it requests a rank comparison from the ACM before presenting the user with information or functionality. As mentioned in the sub-section on the ACM, this is done to ensure that only those with sufficient privileges are allowed access to information and functionality. For example, in Figure 36 in Section 12.2.2 of the previous chapter the 'Remove' button is greyed out. In this case, the GUI module requested a rank comparison from the ACM and determined that the user did not have the right to delete or remove event logs.

In a similar vein to the previous chapter, the next sub-section lists the architectural features that were not implemented by the modules discussed in this chapter.

13.4 Features Not Implemented

In Section 12.3 of the previous chapter we mentioned that the large size of the prototype and a limited amount of available time meant that not all the features of the architecture presented in Chapter 10 could be implemented. To reiterate, non-core functionality, or functionality that could be easily added at a later stage in future work was considered the

lowest priority during development of the prototype. Such functionality may thus have not been included in the prototype. The list below details the features that were omitted in the prototype DFRIMM:

- **Law enforcement contact policy and procedure.** These documents were not added but can be easily added to the 'Documents' menu of the DFRIMM since, in terms of functionality, these documents are no different from any other documents in the DFRIMM.
- **Organisational structure.** This was not implemented; however, if implemented in its most simple form, the organisational structure can be included in the DFRIMM as a document. If it is included as a document, adding the organisational structure is trivial. The organisational structure may also be included in a more elaborate, graphical form, such as an interactive image that users can click on to get information. If a more interactive means is used, significantly more development may be required to add the organisational structure to the DFRIMM.
- **Leave management.** Functionality with which to administer and manage staff leave was not included in the prototype primarily due to the complexity and time required to implement it. Leave management was also not considered core functionality and is therefore left as future work.
- **Investigation archive.** The investigation archive is merely an archive to store files gathered when responding to incidents. The architecture requires that these files are encrypted. Given that encryption libraries are already used in the prototype to encrypt event data, the investigation archive can be implemented by: (1) using existing encryption algorithms to encrypt files to disk or database; and (2) using the existing access control model in the ACM to control access to the files from within the DFRMS. While not trivial, this is straight forward to develop and is also left as future work.

In addition to the list above, we did not implement a costing module in the prototype. The costing module was not developed for two reasons. First, it required a considerable amount of development. Second, the concept of costing in DFR, or more specifically

TDABC applied to DFR, had already been shown through the model and simulations presented in Chapters 8 and 9.

In the next section we conclude this chapter.

13.5 Conclusion

This chapter continued the discussion of the proof-of-concept DFRMS prototype that was introduced in the previous chapter. The majority of the chapter was dedicated to describing the prototype's implementation of the digital forensic readiness information management module (DFRIMM). The discussion of the DFRIMM detailed the functionality for the administration or management of the following: DFRMS users; DF and incident response teams; DFR-related staff training; business process information; and lastly, documentation such as DF-related policies and procedures. In addition to the DFRIMM, the chapter also described the implementation of the access control module and user interface module.

With the exception of the costing module, the prototype implemented all of the modules per the conceptual DFRMS architecture presented in Chapter 10. Some features or functionality of the DFRIMM, namely, law enforcement contact policy and procedure, organisational structure, leave management, and the investigation archive, were not implemented. These were omitted owing to time constraints and a higher priority being placed on the core functionality that was implemented. The costing module was not developed since the concept of TDABC applied to DFR was already shown in the model and simulations discussed in Chapters 8 and 9. The core functionality of the DFRIMM and the other modules implemented by the prototype show that the conceptual DFRMS architecture can be implemented.

Inasmuch as we have shown in both this and the previous chapter that the concept of the DFRMS can be implemented, and implemented according to our architecture, we cannot make an unequivocal claim regarding the feasibility of a DFRMS. There are many human and technical factors that may affect the feasibility of a DFRMS in practice. In order to determine whether a DFRMS is feasible in practice, empirical research in an organisational setting is required. Such research is a significant endeavour that involves

evaluating technical, human and organisational factors. It is therefore beyond the scope of this thesis and we leave it for future research.

The concluding chapter of the thesis follows.